

TITLE OF THE INVENTION

**A 3D SCENE ORIENTATION INDICATOR SYSTEM WITH SCENE ORIENTATION CHANGE
CAPABILITY**

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0001] The present invention is directed to a widget or graphical user interface (GUI) element that depicts three-dimensional (3D) scene orientation and includes direction indicating controls that allow a user to orient a view of the 3D scene to align with the direction indicated by the selected control and, more particularly, to a GUI element that is resident within the 3D scene, stays aligned with the scene as the scene is tumbled by the user, remains in the same position and at the same size in the display as the view changes, and has controls that automatically rotate the scene to a predetermined view when selected where a scene object is centered in the display and sized to fit the display.

2. Description of the Related Art

[0002] In the three-dimensional (3D) graphics art where computers are used to perform animation, 3D modeling and other tasks in which a 3D scene is depicted, it is important to orient the user to the view into the scene and to allow the user to easily change the view into the scene. Conventional interfaces 10 in the 3D art, such as shown in figure 1, can include an orientation gnomon 12 located in a display associated with a 3D scene 14. As the view changes, the gnomon in the display is adjusted to reflect changes in the view into the scene. The gnomon can also be located in the center of the scene or at the origin of the axes of the 3D scene. This gnomon cannot be used to change a view of the 3D scene.

[0003] What is needed is a gnomon that can be used to change a view of a 3D scene.

[0004] When a user wants to change the view of a scene, such as depicted in figure 1, to a particular desired view, the user can manipulate (tumble, rotate, translate, etc.) the scene to get the desired view. However, the user must be particularly aware of the current view and the manipulations needed to get to the desired view. This can be difficult if the view has been tumbled a number of different times along complex tumbling pathways. The user can also change the view to a desired view among a number of predefined views using a menu as depicted in figure 1. Figure 1 depicts a pending selection of a back view. The menu selection

view change process requires that the user move through a number of different menu levels (3 in this example) to make such a selection, which takes time that may be considered wasted. However, even when provided with such a menu, the menu selection needed to get the desired view based on the current view is often not readily apparent especially to an unsophisticated user.

[0005] What is needed is a system that provides the user with orientation information and selectable controls for view manipulation that do not require a complex selection process and that visually guide the user to a desired view with directional aids associated with the selection controls.

SUMMARY OF THE INVENTION

[0006] It is an aspect of the present invention to provide an in scene three-dimensional orientation indicator that automatically remains aligned with the orientation of a three-dimensional (3D) scene.

[0007] It is an additional aspect of the present invention to provide an in-scene indicator that remains positioned at a predetermined position in the display and at a predetermined size in the display as the scene is manipulated into different views by the user.

[0008] It is another aspect of the present invention to provide the indicator with controls or buttons that can be used to change the view of the scene.

[0009] It is also an aspect of the present invention to provide the orientation indicator with direction indicators associated with the controls that indicate the direction of the various views that can be selected.

[0010] It is a further aspect of the present invention to automatically center and fit the objects in the 3D scene to the display view subsequent to a view change.

[0011] The above aspects can be attained by a system that provides an orientation indicator graphical user interface element in a three-dimensional scene where the orientation indicator can be used to automatically change a view of the scene to a predetermined viewpoint. The indicator includes controls that when activated or selected cause the view of the scene to change to a view direction indicated by the control. The view direction associated with a particular control can be indicated by a shape of the control or by the location of a control. The view of the scene is also automatically adjusted at the selected view position to center an object

of interest in the scene in the display view and to fit the object to the display view. The indicator is moved about within the scene and changed in size to position the indicator at a predetermined fixed position in the users display view and at a predetermined fixed size.

[0012] These together with other aspects and advantages which will be subsequently apparent, reside in the details of construction and operation as more fully hereinafter described and claimed, reference being had to the accompanying drawings forming a part hereof, wherein like numerals refer to like parts throughout.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] Figure 1 illustrates a conventional 3D display with a menu for selecting a view of a scene to be displayed.

[0014] Figure 2 depicts a view compass or GUI element according to the present invention.

[0015] Figure 3 shows a view compass in a 3D scene.

[0016] Figures 4-7 illustrate changes in view of a scene and corresponding changes in the view compass.

[0017] Figure 8 illustrates a handle/control of the compass being highlighted as a cursor passes over the handle.

[0018] Figure 9 illustrates the orientation after the handle highlighted in figure 8 has been selected.

[0019] Figure 10 is a flowchart of the process performed when a control of the compass is selected.

[0020] Figures 11-17 illustrate additional embodiments of the present invention.

[0021] Figure 18 illustrates hardware of the present invention

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0022] The present invention is a graphical user interface (GUI) orientation indicator element indicating the orientation of a three-dimensional (3D) scene that has directional buttons or controls that allow the orientation to be changed to align a display view of the scene with the direction indicated by a selected button. The invention can be called a view compass GUI element, tool or widget or just a view compass.

[0023] The view compass 30, as depicted in figure 2, is a collection of seven virtual buttons or

handles 32-44, represented as a 3D figure (widget) on a computer display. The buttons are conventional virtual buttons in that the operating system user interface recognizes that an action is to be taken when a cursor is on one of the buttons and a mouse button is activated or "clicked". The compass 30 becomes or is an element of a 3D scene, such as those used for modeling, CAD, or videogames. These seven buttons or controls 32-44 provide shortcuts for moving the virtual camera in (or viewpoint of) the scene to pre-defined positions.

[0024] The view compass 30 is unique in that its position is coupled to the virtual camera or the point of view into the 3D scene as established by an associated system, such as the Alias ImageStudio™ system, making it always available in the same place. The orientation of the compass 30 is locked to the orientation of the virtual 3D world, so that the action of each view-changing handle is visually obvious. This is a change from current ways of switching views in 3D software, as discussed above, that require the user to mentally calculate (or guess) which absolute position they need to switch to in order to get the view they want.

[0025] The view compass 30 is drawn as a 3D figure, comprising a core handle (or control) 44 with six handles (or controls) 32-42 arranged peripherally around it (one handle on either side in each of the X, Y, and Z directions). The compass 30 preferably has a form, as depicted in figure 2, that allows most or all of the handles to be visible (and therefore pickable) at any given time. For Alias ImageStudio™ we prefer a cubic 'core' 44 with cone-shaped handles 32-42, but many other forms are possible as will be discussed later herein. Preferably, the entire visible portion of a cone is a control.

[0026] The preferred compass 30, as noted above, includes cone shaped handles 32-42 where the point 46 of the cone preferably points in the direction of the virtual camera view point, which provides a visual and somewhat intuitive information about what view will be provided when the handle is selected. The compass 30 preferably also has some additional visual cues. One of the handles, in this case handle 44, is a different color to mark it as the "front" direction of the scene, and the system can label the positive X, Y, and Z axes associated with the handles (see figure 14). These refinements are not required to make the view compass work well, but are preferable additions.

[0027] There are a number of aspects of the view compass 30 that it is important to understand. The first is how the compass 30 it behaves in the scene since it is part of the scene. As the scene view is changed the compass 30 changes with the scene. For example, as the scene is tumbled, because the compass is in the scene, the compass 30 also tumbles with the scene

automatically maintaining an orientation that matches the orientation of the scene. However, as the scene is tumbled, the compass would normally move within the display view or virtual camera view of the scene provided to the user. As another example, because the compass is part of the scene, if the user zooms into the scene, the size as well as the position of the view compass 30 would be expected to change in the display image or display view presented to the user. The present invention preferably maintains the compass in the same position (preferably a display corner) and at the same size, as will be discussed in more detail below. A second aspect is how the displayed scene changes when a handle of the compass is selected. When a handle is activated the scene is switched to the viewpoint of the handle, the 3D scene object is centered and the display view is resized (zoom-in or zoom-out) so that the object fits within the display as will be discussed in more detail later herein.

[0028] At all times, the view compass 62 is preferably drawn in the view 64 that it controls. Its position and size are fixed relative to the virtual camera (and in terms of the number of pixels on the computer display, preferably about 110 by 110 pixels), but the compass 62 orientation is fixed relative to the 3D world coordinate system, as can be seen in figure 3. This way, as the camera moves, the view compass 62 stays constantly at one size, in one out-of-the-way corner of the screen (upper right in figure 3). In this position, it also tumbles to match the orientation of the virtual world (note that the front position points along the axis of the ring in figure 3). When the computer redraws the scene, the view compass 62 is preferably drawn last so that it is drawn on top of everything else in the scene. This ensures that the compass 62 is always visible, even if scene objects come between the compass 62 and the virtual camera.

(Depending on the details of the particular graphics system being used, it may be necessary to turn off the z-buffer, or ensure that the compass is just past the near clipping plane.)

[0029] Set forth below is pseudo-code for an algorithm that can be implemented in a preferred graphic system in a language, such as C++, and that places the view compass 62 at a fixed position and size, while allowing it to rotate with the world axes.

```

// Information required for this procedure:
//
point cameraPosition; // 3d position of the virtual camera
vector cameraDirection; // normalized vector in the direction of view
vector cameraUp; // normalized vector pointing "up" in camera-space
vector cameraRight; // normalized vector pointing "right" in camera-space
angle viewAngle; // horizontal angle of the view frustum of this camera

```

```

angle viewAngle; // vertical angle of the view frustum of this camera

int viewportWidth; // width of the screen display in pixels
int viewportHeight; // height of the screen display in pixels

const int compassSize; // size of the view compass in pixels
const int compassOffset; // offset of the view compass center
                        // from the corner in pixels

procedure drawViewCompass( camera, viewport ){

    // calculate the width/height of the viewing frustum at
    // one unit in front of the camera
    //
    float frustumWidth = 2.0 * sin(viewAngle*0.5)/cos(viewAngle*0.5);
    float frustumHeight = 2.0 * sin(viewAnglev*0.5)/cos(viewAnglev*0.5);

    // calculate the width of a pixel at a distance one unit in front
    // of the camera
    //
    float pixelSize = (frustumWidth/viewportWidth);

    // calculate the size of the compass in real-world units
    //
    setCompassSize( compassSize * pixelSize );

    // calculate the position of the compass center
    //
    setCompassPosition( cameraPosition + cameraDirection +
        ((frustumWidth*0.5)-(pixelSize * compassOffset)) * cameraRight +
        ((frustumHeight*0.5)-(pixelSize * compassOffset)) * cameraUp );

    // No need to set the orientation; it's always aligned with the
    // world.  We can now draw the compass at the appropriate position,
    // size, and orientation.

    ...
}

```

[0030] The process set forth above starts by calculating how big the view compass would have to be in 3D space to maintain the same apparent size in pixels on the current display. This can change based on the focal length of the virtual camera, or on the size of the window in which the program is running. Next, the process calculates where in 3D space the view compass would need to be located in order to maintain the same apparent position relative to the camera. Once that is done, the compass widget itself can be drawn in a conventional way.

[0031] The process discussed above is executed each time the user changes the viewpoint of the scene, such as by tumbling, rotating, translating, etc. the scene. This is depicted in figures 4-7, which show a user tumbling the view from a perspective view (figure 4) to an offset top view (figure 5), then to an upside down view (figure 6) and finally to a left side view (figure 7). In figure 4, the front cone of the compass 62 is pointing perspectively in alignment with the axis of the ring scene object 66 in the scene 64. Figure 5 shows the compass 62 also in the same corner and the same size as in figure 4, even though the ring appears to have become a bit larger while the compass 62 has also been tumbled to keep the front cone aligned with the ring axis and in the relative front position of the ring 66. Figure 6 again shows the compass 62 in the upper right hand corner at the same size and tumbled to be in alignment with the scene 64. Figure 7 depicts the view compass 62 in a left side view aligned with the ring 66, so that the front cone points in the axial direction of the ring 66.

[0032] In addition, the above discussed compass move and resizing process is executed each time the user changes the viewpoint of the scene by selecting one of the controls of the view compass as discussed below.

[0033] As discussed above, each handle of the view compass 72 acts as a button or control. As a cursor passes over each handle, as depicted in figure 8, the handle is conventionally highlighted in the scene 74 to show that the handle will be picked if the user clicks-on or selects the handle. When the user clicks-on one of the cone handles, the camera moves to align its view direction with the axis of the cone as depicted in figure 9. (That is, the camera turns around the scene to point the same way that the cone points.) After the click in figure 8, the camera is now looking along the same axis as the clicked handle, down the center of the ring 76 in the scene 74, and the screen is filled with the object of interest (the ring 76 in this example) as shown in figure 9. The view compass 72 has the same relative position and size in the scene 74, but has rotated to a new orientation, matching the worldview. Once in selected orientation, the virtual camera or display view is repositioned so that objects of interest in the scene are centered and made to fit into the view, as shown in figure 9. The objects of interest depend on the application. For example, the view compass could focus on selected objects, or on all objects. The camera, as shown in figure 9, also switches from a perspective to an orthographic mode. When the user clicks-on or selects the center handle or core (see 44 of figure 2) of the view compass, the camera rolls to a predefined view such as the a predefined perspective view, such as the home view depicted in figure 8. Again, the camera adjusts its

position to ensure that all objects of interest are made to fit into the view. The operations of the software that controls how the compass behaves will be discussed in more detail below with respect to figure 10.

[0034] The flowchart of figure 10, for convenience, depicts a process 90 that is sequential. However, when implemented in an object-oriented language, such as the preferred C++ language, the tasks may be distributed into objects that may not be sequentially organized. Before any selection is made, a conventional operation (not shown) highlights any handle over which the cursor passes and the process waits 92 for a selection. Where a selection is preferably first processed by the compass behavior software described herein. If the cursor is used to make a selection, such as by clicking a button on a mouse or engaging a key on a keyboard, the process leaves the wait state 92, and a determination 94 is made as to whether the selection, cursor click-hit, is a selection of a control of the view compass. If not, the selection event is passed 96 to other software of the system. If so, a determination 98 is made as to which handle or control has been hit. If the top handle is selected (hit), the view or virtual camera is conventionally rolled or tumbled 100 to display the top view to the user. If the bottom handle is selected (hit), the view is conventionally rolled 102 to display the bottom view. Likewise for the left (104), right (106), front (108) and back (110) views. When the home or core control is selected, the view provided to the user is rolled 112 to the perspective view. The system then conventionally centers 114 the object of interest and fits the display size to the object as previously discussed. Next, the view compass size and position are adjusted 116 in the 3D scene to place the compass in the desired predetermined position at the predetermined size. The system then returns to await another input event.

[0035] The view compass of the present invention can have a number of different shapes or other features that allow the direction of the view to be visualized, that depict the orientation and that are easy to select. The cone shaped controls of figure 2 can be pyramid shaped. Arrows can be substituted for the cones. Figure 11 depicts a compass 130 with axial indicators 132, 134 and 136 for the xyz axes. Figure 12 illustrates a compass 150 where balls 152 are the controls and the links 154 between the balls 152 indicate view direction. That is, the stick of the ball and stick combination points at the core indicating view direction. Figure 13 shows a compass 160 where the views are indicated by planes or windows 162. The direction of the view is indicated by the location of the window and by the view through the window to the core. Figure 15 shows a compass 180 with view bookmarks 182, 184 and 186 that indicate additional

system or user created views to which the system can align the scene when selected. These additional views can be used for special purposes, such as in modeling where people often want to "bookmark" certain views, that is, save them because they contain an attractive angle for rendering, or because they are focused on a particular detail that they want to work with, and need to get back to periodically. Or, the user may be saving views of particular interest to point them out to a colleague. Additional components or scene views markers associated with the view compass can be provided as depicted in figure 16. In this embodiment, the view compass 200 has additional view markers, such as 202, 204 and 206, arbitrarily located at segment boundaries of a grid plane and that cause the display view to align with the particular plane of the view marker when selected. The view markers of the present invention can also be exploded as depicted in figure 17 where the display view 220 is of the entire 3D scene 222 and the markers or controls, such as controls 224, 226 and 228 are located at the periphery of the scene.

[0036] Figure 18 illustrates hardware of the present invention where the 3D scene, including the view compass, is depicted on a display 250, such as a CRT, PDP or LCD. A computer 252, such as desk top computer, runs the processes of the present invention, along with the 3D graphics software, that allow the user to manipulate the scene and select controls on the display, via input devices, such as a mouse 256 and keyboard 258. Other input devices can also be used, such as tablets, gloves, etc., that allow a user to point at and select objects.

[0037] The system of the present invention also includes permanent or removable storage, such as magnetic and optical discs, RAM, ROM, etc. on which the process and data structures of the present invention can be stored and distributed. The processes can also be distributed via, for example, downloading over a network such as the Internet.

[0038] The present invention has been described with the visible part of each cone being a control. It is possible to have a region around each cone that also acts as a control. The axis labels (x, y, z) can also be part of the controls. The present invention has been described as making a front cone a different color to indicate a front direction of the compass and scene. It is possible to make a front face of the core 44 a different color either in addition to or as a substitute for the coloring of the front cone. Iconic representations of the contents of the scene from the various points of view can also be placed on the core faces.

[0039] It is also possible to use the compass as a tumbling tool such that dragging a cone will turn the scene. It is also possible to drag an image, such as a texture, and drop it on a cone

and have the system paste the image on a part of the scene such as a wall.

[0040] The many features and advantages of the invention are apparent from the detailed specification and, thus, it is intended by the appended claims to cover all such features and advantages of the invention that fall within the true spirit and scope of the invention. Further, since numerous modifications and changes will readily occur to those skilled in the art, it is not desired to limit the invention to the exact construction and operation illustrated and described, and accordingly all suitable modifications and equivalents may be resorted to, falling within the scope of the invention.